J. Symbolic Computation (1990) **9**, 355–373

Computing the Structure of Finite Algebras

LAJOS RÓNYAI†

Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, P.O.B. 63, *H*-1502 *Hungary*

(*Received* 1 *October* 1987)

In this paper we address some algorithmic problems related to computations in finite-dimensional associative algebras over finite fields. Our starting point is the structure theory of finite-dimensional associative algebras. This theory determines, mostly in a nonconstructive way, the building blocks of these algebras. Our aim is to give polynomial time algorithms to find these building blocks, the radical and the simple direct summands of the radical-free part. The radical algorithm is based on a new, tractable characterisation of the radical. The algorithm for decomposition of semisimple algebras into simple ideals involves (and generalises) factoring polynomials over the ground field.

Next, we study the problem of finding zero divisors in finite algebras. We show that this problem is in the same complexity class as the problem of factoring polynomials over finite fields. Applications include a polynomial time Las Vegas method to find a common invariant subspace of a set of linear transformations as well as an explicit isomorphism between a given finite simple algebra and a full matrix algebra over a finite field.

1. Introduction

Our main objects of study are associative algebras. A is an *associative algebra* over the field F if:

(a) A is a linear space over F;
(b) A is equipped with a binary F-bilinear operation $*$ (i.e. the multiplication);
(c) the multiplication is associative:

$$x * (y * z) = (x * y) * z \text{ holds for every } x, y, z \in A.$$

As is usual, we write xy instead of $x * y$. In this paper we shall consider finite-dimensional algebras only, i.e. we assume that $dim_F A$ is finite. We shall use the terms and notions *commutative algebra*, *subalgebra*, *(left) ideal*, *factor algebra*, *homomorphism*, *A-module* in the standard way, cf. Herstein (1968) and Pierce (1982).

An algebra A is *simple* if A has only trivial ideals (i.e. (0) and A) and $AA \neq (0)$. We say that A is the direct sum of its (left) ideals $A_1, \ldots, A_k$ (written as $A_1 \oplus \cdots \oplus A_k$) if A is the direct sum of these linear subspaces.

EXAMPLES.

(a) If the field K is a finite algebraic extension of the field F then K is a finite-dimensional simple and commutative algebra over F.
(b) $M_n(F)$, the algebra of all n by n matrices over the field F. $M_n(F)$ is a simple algebra over F and $dim_F M_n(F) = n^2$.

† Research partially supported by Hungarian National Foundation for Scientific Research, Grant 1812.

0747–7171/90/030355+19 $03.00/0

(c) Subalgebras of $M_n(F)$, i.e. linear subspaces of $M_n(F)$ closed under matrix multiplication.

The latter examples tend to be typical as the following well-known representation theorem shows.

REPRESENTATION THEOREM. *Let A be an algebra over the field F and suppose that $\dim_F A = n$. Then A is isomorphic to a subalgebra of $M_{n+1}(F)$. Moreover, if A has an identity element, then A is isomorphic to a subalgebra of $M_n(F)$.*

If A has no identity element then we can adjoin one using the Dorroh extension (cf. Kertész, 1987, p. 43) with the ground field. This process increases the dimension of A by one.

The Representation Theorem is easily proved using the *regular representation* of A. For each $x \in A$ we define the linear map $R_x: A \to A$ as $R_x(y) = xy$ for every $y \in A$. It is easy to see that R is an algebra homomorphism of A to the algebra of linear transformations of the linear space A. If A has an identity element then R is injective.

An element $x \in A$ is *nilpotent* if $x^m = 0$ for some positive integer exponent m. An element $x \in A$ is *strongly nilpotent* if xy is nilpotent for every $y \in A$. The *radical* of A, denoted by $Rad(A)$, is defined as

$$Rad(A) = \{x \in A;\ x \text{ is strongly nilpotent}\}.$$

It is not difficult to see that $Rad(A)$ is an ideal of A and the factor algebra $A/Rad(A)$ has no strongly nilpotent elements except 0. Algebras having no strongly nilpotent elements except 0 are called *semisimple algebras*. Semisimple algebras admit a very nice structure theorem due to Wedderburn (cf. Herstein, 1968; Pierce, 1982).

WEDDERBURN STRUCTURE THEOREM. *Suppose that A is a finite-dimensional semisimple algebra over the field F. Then A can be expressed as a direct sum of simple algebras*

$$A = A_1 \oplus A_2 \oplus \cdots \oplus A_k, \tag{1.1}$$

where the A_i are the only minimal nontrivial ideals of A. Moreover, each A_i is isomorphic to a full matrix algebra $M_{n_i}(F_i)$ where F_i is a not necessarily commutative extension field of F.

We remark that the theorem of Wedderburn on finite division rings (cf. Herstein, 1968, p. 71) implies that if F is finite then the fields F_i are commutative.

We need some facts on Lie algebras. The results quoted can be found in Jacobson (1962) and Humphreys (1980). A *Lie algebra* over the field F is a linear space L over F equipped with an F-bilinear binary operation $[\ ]$ such that the following identities are valid for any $x, y, z \in L$.

(a) $[xx] = 0$,
(b) $[[xy]z] + [[yz]x] + [[zx]y] = 0$.

As in the associative case, one can speak about *subalgebras, ideals, factor algebras* and *homomorphisms* of Lie algebras. The *derived series* of L is the sequence $L^{(i)}$ of ideals of L where $L^{(0)} = L$ and $L^{(i+1)} = [L^{(i)}L^{(i)}]$ for $i > 0$. L is *solvable* if $L^{(n)} = (0)$ for some natural number n. If L is finite dimensional over F then it has a unique maximal solvable ideal $R(L)$, the *radical* of L.

The *descending central series* of L is the sequence L^i of ideals of L where $L^0 = L$ and $L^{i+1} = [LL^i]$ for $i > 0$. L is *nilpotent* if $L^n = (0)$ for some natural number n. If L is finite dimensional over F then it has a unique maximal nilpotent ideal $N(L)$, the *nilradical* of L.

EXAMPLE. For $A, B \in M_n(F)$ let $[AB] = AB - BA$ (i.e. the additive commutator). It is easy to see that this operation is F-bilinear and satisfies requirements (a) and (b), so if a subspace L is closed under the operation [ ] then L can be considered as a Lie algebra. Lie algebras obtained in this way are called *linear Lie algebras.*

There is a straightforward but usually not faithful representation of abstract Lie algebras as linear Lie algebras. For an $x \in L$ let $ad(x): L \to L$ be the linear map for which $ad(x)y = [xy]$. The map $x \mapsto ad(x)$ is a Lie algebra homomorphism of L to the linear Lie algebra of all linear transformations of L. We remark that a deep theorem of Iwasawa and Ado (Jacobson, 1962, chap. 6) states that every finite-dimensional Lie algebra is actually isomorphic to a linear Lie algebra.

When speaking of algorithms, one has to specify the input of the algorithmic questions considered. An algebra (associative or Lie) can be given by a collection of *structure constants.* If A is an algebra over a field F and $a_1, a_2, \ldots, a_n$ is a linear basis of A then multiplication $\circ$ can be specified by representing the products $a_i \circ a_j$ as linear combinations of the a_i

$$a_i \circ a_j = \gamma_{ij1} a_1 + \cdots + \gamma_{ijn} a_n.$$

The coefficients $\gamma_{ijk} \in F$ are called structure constants. In this paper algebras are considered to be given as a collection of structure constants. As a special case, F is an algebra over its prime field, therefore F can also be inputted with structure constants. Finite fields can be (and often are) specified by giving the minimal polynomial f of a single generating element α over the prime field $GF(p)$. Notice that this representation is a special case of the representation with structure constants. The coefficients of f give the structure constants with respect to the basis $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$ of F where $n = dim_{GF(p)} F$.

We remark that algebras may be given as matrix algebras. In this case it suffices to specify a set of matrices which generates the algebra. Our results are applicable in this setting as well, since from this representation one can efficiently find a basis of the algebra and then the structure constants with respect to this basis. In the case of associative algebras the regular representation gives an efficient method to obtain a matrix representation from structure constants. We mention here that the methods do not seem to be directly applicable when the algebras are given by generators and relations.

Our aim is to give algorithms which run in time polynomial in terms of the input size (cf. Hopcroft & Ullman, 1979, Chapters 12–13). Modulo p residue classes have size $\lceil \log_2(p+1) \rceil$. The size of composite objects (matrices, vectors, etc.) can be obtained by adding up the sizes of their parts. Polynomial time algorithms are known (Knuth, 1981; Lidl & Niederreiter, 1983) for the basic seminumerical problems (such as arithmetical operations in finite fields, polynomial arithmetic over finite fields). Also, we have efficient methods to solve systems of linear equations over finite fields. A central algorithmic problem in this context is the problem of factoring polynomials over finite fields. A deterministic method was given by Berlekamp (1968; see also Lidl & Niederreiter, 1983). The time complexity of this algorithm is a polynomial in the parameters p, s and $deg(f)$ where $f \in GF(q)[x]$ is the polynomial to be factored and $q = p^s$, p prime. Note that the input size in this case is $O((1 + deg(f)) \log q)$, consequently the running time of the above method is not polynomial in the input size. The problem can be solved in polynomial time if we

allow randomisation. The first such method was given by Berlekamp (1970) [see also Ben Or (1981); Camion (1983); Cantor & Zassenhaus (1981); Rabin (1980)]. Actually, this method belongs to a special kind of randomised algorithms. For an arbitrary input it either gives a correct solution or, with a small probability, reports failure (i.e. never gives an incorrect answer). Such methods are called *Las Vegas algorithms*. The term was introduced by Babai (1979). We will use both deterministic and Las Vegas factoring algorithms. To handle this situation, the following definition will be convenient. By an *f-algorithm* we mean an algorithm which uses an oracle (subroutine) to factor polynomials over finite fields. The cost of a call of this oracle is the length of the input of the call.

In section 2 we give a deterministic polynomial time algorithm to compute the radical of finite algebras. This method will depend on a new, algorithmic characterisation of the radical. This result is then applied to the computation of the (solvable) radical and the nilradical of finite Lie algebras.

In section 3 a polynomial time f-algorithm is described to find the Wedderburn decomposition (1.1) of a finite semisimple algebra A. The algorithm is based on the work of Friedl (1983), cf. Friedl & Rónyai (1985).

In section 4 a polynomial time f-algorithm is given to find zero divisors in a finite algebra A, i.e. nonzero elements $x, y \in A$ such that $xy = 0$. It may come as a surprise that the case $A \cong M_n(F)$ presents most of the difficulties here. The main algorithm of section 3 can be viewed as a special case of the zero divisor algorithm.

In section 5 we give some applications of the zero divisor algorithm. First, we develop an important auxiliary procedure to find explicit isomorphisms of full matrix algebras. This enables us to find a common invariant subspace for a set of linear operators and to express a finite module over a finite semisimple algebra as a direct sum of simple submodules. These algorithms also run in polynomial time, as f-algorithms. Finally, we derive a deterministic polynomial time procedure to solve the following computational problem on permutation groups, raised by W. M. Kantor.

Let $G \leq S_n$ and let $K < H$ be normal subgroups of G such that H/K is an elementary Abelian p-group for some prime p. The groups G, H and K are given by generating sets. Find a normal subgroup L of G, minimal subject to the conditions $K < L \leq H$.

Our primary objective is to exhibit (deterministic or randomised) polynomial time algorithms to solve the problems. Beyond that we do not address the efficiency of the methods. In fact, in many cases these results can only be regarded as first steps towards obtaining good algorithms.

2. Computing the Radical

In this section we give a polynomial time algorithm to compute the radical of finite associative algebras. The method can be applied to the problem of computing the nilradical and the solvable radical of finite Lie algebras. The main algorithmic problem we consider here is the following.

Given a finite-dimensional associative algebra A over the field $GF(q)$ by a collection of structure constants, find a basis of $Rad(A)$, the radical of A in time polynomial in the input size, i.e. the parameters $dim_{GF(q)}A$ and $\log q$.

We claim that it suffices to solve the problem for algebras over a prime field $GF(p)$. Indeed, A can be considered as an algebra over $GF(p)$ and the radical, as the set of strongly nilpotent elements of A, does not depend on the ground field considered. Also, this change

does not affect the input size. Finally, from a $GF(p)$ basis of $Rad(A)$ we can easily obtain a basis over $GF(q)$.

We remark that the problem of finding the radical over fields of characteristic zero is equivalent to solving a system of linear equations over the ground field. One can use the following theorem of Dickson (1923, pp. 106–108) on the radical of matrix algebras.

DICKSON'S THEOREM. *Let A be a finite-dimensional algebra of matrices over a field F such that $charF = 0$. Then*

$$Rad(A) = \{x \in A;\ Tr(xy) = 0 \text{ for every } y \in A\}.$$

This result shows that if $charF = 0$ then $Rad(A)$ can be obtained by solving a system of linear equations over F.

EXAMPLE. Let A denote the algebra of all p by p scalar matrices over the finite prime field $GF(p)$. Clearly, A is semisimple but the trace form identically vanishes on A. This shows that the condition on the characteristic of the field in Dickson's theorem cannot be dropped.

We return to the finite case. Let p be a prime and let $F = GF(p)$ and suppose that A is a subalgebra of $M_n(F)$ where $dim_F A = n$ or $dim_F A = n-1$. Using the regular representation of A, we can efficiently achieve this situation. We define the natural number l by the following inequalities: $p^l \leq n < p^{l+1}$. Let B denote the set of matrices $A \cup \{1_n\}$ where 1_n is the identity element of $M_n(F)$.

Our main objective is to define a sequence of ideals $I_{-1}, I_0, \ldots, I_l$ of A and a sequence of functions $g_i: I_{i-1} \to F$, $i = 0, 1, \ldots, l$ with the following properties.

(1) $I_{-1} = A$ and $I_l = Rad(A)$.
(2) g_i is an F-linear function for $i = 0, \ldots, l$.
(3) $I_i = \{x \in I_{i-1};\ g_i(xy) = 0 \text{ for every } y \in B\}$.
(4) $g_i(x)$ can be computed for any $x \in A$ in time polynomial in n and $\log p$.

These properties immediately imply that from a basis of I_{i-1} we can obtain a basis of I_i by solving a system of linear equations over F. By linearity of the functions g_i and by property (4) the coefficients of this sytem can be obtained in polynomial time. If we start with $I_{-1} = A$ then we can find $Rad(A)$ using $l+1 = O(\log n)$ iterations, therefore (a basis of) $Rad(A)$ over F can be obtained using $(n + \log p)^{O(1)}$ bit operations.

Before defining these ideals and linear functions we need some preparation.

Let M_n denote the ring of n by n matrices over the integers and ϕ denote the ring homomorphism from M_n to $M_n(F)$ induced by the $Z \to F$ epimorphism (i.e. ϕ denotes the *mod p* reduction of integral matrices). Matrices over Z will be denoted by capitals (C, D, X, Y), the corresponding lower-case letters denote matrices over F.

We want to speak about $Tr(c^{p^i})(mod\ p^{i+1})$ where $c \in M_n(F)$ by simply choosing an arbitrary integral matrix $C \in M_n$ for which $\phi(C) = c$ and taking $Tr(C^{p^i})(mod\ p^{i+1})$. This procedure is justified by the following lemma.

LEMMA 2.1. *Let $C, D \in M_n$ and suppose that $\phi(C) = \phi(D)$. Then for any natural number i we have*

$$Tr(C^{p^i}) \equiv Tr(D^{p^i})(mod\ p^{i+1}).$$

PROOF. The statement is trivial for $i = 0$, therefore we may assume that $i > 0$. Let

$P = D - C$. It is clear that every entry of P is divisible by p. First we notice that if $B_1, \ldots, B_k$ are integral matrices and m of them are equal to P, then every entry of the product matrix $B = B_1 \cdots B_k$ is divisible by p^m, therefore $Tr(B)$ is also divisible by p^m.

We expand the right-hand side of the congruence stated. We obtain

$$Tr(D^{p^i}) = Tr((C+P)^{p^i}) = \sum Tr(Z_1 Z_2 \cdots Z_{p^i}),$$

where Z_j is either C or P and the summation ranges over all such products. Now let $G = \langle \pi \rangle$ denote the cyclic group of order p^i. We define an action of G on the words $Z_1 \cdots Z_{p^i}$ by setting

$$\pi(Z_1 Z_2 \cdots Z_{p^i}) = Z_{p^i} Z_1 Z_2 \cdots Z_{p^i - 1},$$

i.e. π acts as a cyclic shift. Clearly, if V and W are two words from the same G-orbit then for the corresponding products (denoted by the same letters) we have $Tr(V) = Tr(W)$ because of the identity $Tr(XY) = Tr(YX)$, $X, Y \in M_n$. If the orbit of the product V contains p^j elements then the contribution of this orbit to the sum is $p^j Tr(V)$. In this case π^{p^i} leaves V fixed, therefore V can be obtained as the p^{i-j}th power of its first p^j factors.

If V as a word is not C^{p^i} then at least p^{i-j} of the matrices appearing in the word V must be equal to P. Now using the trivial inequality $p^{i-j} \geq i-j+1$ we obtain that every entry and hence the trace of V is divisible by p^{i-j+1}, therefore the contribution of the orbit is divisible by p^{i+1}. Finally, we observe that the word C^{p^i} forms a one element orbit, proving the lemma. □

The next lemma provides a tool for inductive arguments.

LEMMA 2.2. *Let H be a multiplicatively closed subset of M_n and let k be a positive integer. Suppose that $Tr(X^{p^i})$ is divisible by p^{i+1} for every $X \in H$ and $0 \leq i < k$. Then we have*

$$Tr((X+Y)^{p^k}) \equiv Tr(X^{p^k}) + Tr(Y^{p^k}) \ (mod\ p^{k+1})$$

for every $X, Y \in H$.

PROOF. This time we expand the left-hand side of the congruence. We obtain

$$Tr((X+Y)^{p^k}) = \sum Tr(Z_1 Z_2 \cdots Z_{p^k}),$$

where Z_j is either X or Y and the summation ranges over all of the 2^{p^k} such products. Let $G = \langle \pi \rangle$ denote the cyclic group of order p^k. Again, the elements of G act on the words $Z_1 Z_2 \cdots Z_{p^k}$ as cyclic shifts. If the orbit of the product V contains p^j elements then the contribution of this orbit to the sum is $p^j Tr(V)$. In this case, as in Lemma 2.1, the matrix V can be obtained as the p^{k-j}th power of its first p^j factors. This prefix is denoted by U. Clearly, we have $U \in H$ and if $j \neq 0$ then our assumptions imply $Tr(V) \equiv 0 (mod\ p^{k-j+1})$. The sum of the orbit of V is divisible by p^{k+1}. The one element orbits of G correspond to the right-hand side of the congruence stated. □

Let $f \in F[x]$ be a monic polynomial:

$$f(x) = x^n + a_1 x^{n-1} + \cdots + a_n. \qquad (*)$$

Let $\alpha_1, \ldots, \alpha_n$ be the roots of f (in a suitable extension of F) and put

$$s_i = \alpha_1^i + \alpha_2^i + \cdots + \alpha_n^i, \quad i = 1, 2, \ldots, n.$$

The elements s_i can be expressed in terms of the coefficients of f, using Newton's identities

$$\begin{aligned} s_1 + a_1 &= 0 \\ s_2 + a_1 s_1 + 2a_2 &= 0 \\ &\vdots \\ s_n + a_1 s_{n-1} + \cdots + a_{n-1} s_1 + n a_n &= 0. \end{aligned}$$

Using these formulae we can establish a trace condition for nilpotence. We recall that l is the unique integer determined by the inequalities $p^l \leq n < p^{l+1}$.

LEMMA 2.3. *Let H be a multiplicatively closed subset of M_n and suppose that $Tr(X^{p^l}) \equiv 0 (mod\ p^{l+1})$ for every $X \in H$. Then $\phi(X)$ is nilpotent for every $X \in H$.*

PROOF. It suffices to show that $\phi(X)^{p^l} = \phi(X^{p^l})$ is nilpotent. Let

$$f(x) = x^n + a_1 x^{n-1} + \cdots + a_n$$

denote the characteristic polynomial (with leading coefficient 1) of the matrix $Y = X^{p^l}$ over the rationals. Clearly, $\phi(Y)$ is nilpotent if and only if $a_i \equiv 0 (mod\ p)$ for $1 \leq i \leq n$. We shall use Newton's identities for the polynomial f. Using the facts that $s_i = Tr(Y^i) = Tr((X^i)^{p^l})$ and that $X^i \in H$, we obtain that s_i is divisible by p^{l+1}. Now from Newton's identities we see that $i a_i \equiv 0 (mod\ p^{l+1})$ for $1 \leq i \leq n$. The definition of l implies that i is not divisible by p^{l+1}, hence a_i is divisible by p for $1 \leq i \leq n$ and the statement follows. □

Next, we prove a counterpart of Lemma 2.3, a necessary condition for nilpotence.

LEMMA 2.4. *Let $X \in M_n$ be a matrix such that $\phi(X)$ is nilpotent. Then*

$$Tr(X^{p^i}) \equiv 0 (mod\ p^{i+1})$$

holds for every natural number i.

PROOF. As $\phi(X)$ is nilpotent, it is similar over F to a strictly upper triangular matrix. Or, expressing this in terms of integral matrices, there exist $C, D, P, R, U \in M_n$ such that

$$CXD = U + P, \quad DC = I + R, \quad U^n = 0 \quad \text{and} \quad \phi(P) = \phi(R) = 0,$$

where I is the identity matrix (in M_n) and 0 denotes the zero matrix of M_n and $M_n(F)$ as well.

Using Lemma 2.1 we have

$$0 \equiv Tr(U^{p^i}) \equiv Tr((U+P)^{p^i}) = Tr((CXD)^{p^i}),$$

where the congruence is *mod* p^{i+1}. We have also

$$Tr((CXD)^{p^i}) = Tr((DCX)^{p^i}) = Tr((X + RX)^{p^i}).$$

Observing that $\phi(RX) = 0$, we can use Lemma 2.1 again

$$Tr((X+RX)^{p^i}) \equiv Tr(X^{p^i}) (mod\ p^{i+1}).$$

By combining these equalities and congruences we obtain that $Tr(X^{p^i})$ is divisible p^{i+1}. □

After these preparations we can define the ideals I_j of the matrix algebra $A \leq M_n(F)$. Let

$I_{-1} = A$ and for $0 \le i \le l$ let

$$I_i = \{x \in A;\ Tr((xy)^{p^j}) \equiv 0 (mod\ p^{j+1}) \text{ for every } y \in B \text{ and } 0 \le j \le i\}.$$

Here, $B = A \cup \{I\}$, where I is the identity matrix of $M_n(F)$ and $Tr(a^{p^j})(mod\ p^{j+1})$ is defined for $a \in M_n(F)$ by choosing an integral matrix $A \in M_n$ for which $\phi(A) = a$ and taking $Tr(A^{p^i})(mod\ p^{i+1})$. This procedure is justified by Lemma 2.1.

From the definition it is immediate that

$$A = I_{-1} \supseteq I_0 \supseteq \cdots \supseteq I_l.$$

Now we can show that these subsets are actually ideals of A.

THEOREM 2.5. *I_m is an ideal for A for $m = -1, 0, 1, \ldots, l$. Moreover, we have $I_l = Rad(A)$.*

PROOF. The first statement is obvious for $m = -1$, so we may suppose that $m \ge 0$. The definition of I_m immediately implies that if $x \in I_m$ and $u \in A$ then $xu \in I_m$. The relation $ux \in I_m$ is obtained from the following identity

$$Tr(((UX)Y)^k) = Tr((X(YU))^k), \quad U, X, Y \in M_n,$$

where k is a nonnegative integer. We have established that I_m is a semigroup ideal of A. Now we show that I_m is an additive subgroup of A. This is clear for $m = 0$ because Tr is an additive function. Now let $m > 0$. As I_m is multiplicatively closed, this is true for its preimage $J_m = \phi^{-1}(I_m)$. We shall apply Lemma 2.2 with $H = J_m$. Let $X, Y \in J_m$ and let $U \in M_n$ such that $\phi(U) \in B$. Now for $0 \le k \le m$ we have

$$\begin{aligned} Tr(((X+Y)U)^{p^k}) &= Tr((XU+YU)^{p^k}) \\ &\equiv Tr((XU)^{p^k}) + Tr((YU)^{p^k}) \equiv 0\ (mod\ p^{k+1}), \end{aligned}$$

where the first congruence follows from the additivity of Tr for $k = 0$ and from Lemma 2.2 for $k > 0$. As for the last congruence, $\phi(X), \phi(Y) \in I_m$ imply that

$$Tr((XU)^{p^k}) \equiv Tr((YU)^{p^k}) \equiv 0\ (mod\ p^{k+1}).$$

We conclude that I_m is an additive subgroup, therefore an ideal of A.

Finally, we show that $I_l = Rad(A)$. Indeed, if $x \in Rad(A)$ then xy is nilpotent for every $y \in B$. Let U be an integral matrix such that $\phi(U) = xy$. Then by Lemma 2.4 we have

$$Tr((xy)^{p^i}) \equiv Tr(U^{p^i}) \equiv 0\ (mod\ p^{i+1}),$$

where i is an arbitrary natural number, showing that $x \in I_l$. It remains to see that $I_l \le Rad(A)$. This immediately follows from Lemma 2.3 if we put $H = \phi^{-1}(I_l)$. The proof is complete. □

For $0 \le i \le l$ we define the functions $f_i : M_n \to Q$ as follows

$$f_i(X) = \frac{Tr(X^{p^i})}{p^i}.$$

If $X \in \phi^{-1}(I_{i-1})$ then $f_i(X)$ is an integer and if $X, Y \in \phi^{-1}(I_{i-1})$ then we have

$$f_i(X+Y) \equiv f_i(X) + f_i(Y)\ (mod\ p). \tag{2.1}$$

The congruence (2.1) is immediate for $i = 0$ and follows from Lemma 2.2 for $i > 0$. Next, we

define the functions $g_i: I_{i-1} \to F$, $i = 0, \ldots, l$ as

$$g_i(x) = f_i(X) \ (mod\ p),$$

where X is an arbitrary integral matrix for which $\phi(X) = x$. To justify this definition, let $X, Y \in M_n$ be such that $\phi(X) = \phi(Y) = x$. Then by Lemma 2.1

$$Tr(X^{p^i}) \equiv Tr(Y^{p^i}) \ (mod\ p^{i+1}),$$

$X, Y \in \phi^{-1}(I_{i-1})$ implies that p^i divides both sides of the congruence, therefore

$$f_i(X) \equiv f_i(Y) \ (mod\ p)$$

follows.

The main properties of the functions g_i are expressed in the following.

THEOREM 2.6. *For $i = 0, 1, \ldots, l$ we have*

(i) $g_i: I_{i-1} \to F$ is an F-linear function;
(ii) $I_i = \{x \in I_{i-1};\ g_i(xy) = 0$ for every $y \in B\}$;
(iii) For a given $x \in I_{i-1}$, $g_i(x)$ can be computed using $(\log p + n)^{O(1)}$ bit operations.

PROOF.

(i) Is an immediate consequence of (2.1);
(ii) this statement is a simple reformulation of the definition of I_i. Indeed, $g_i(xy) = 0$ if and only if $Tr((xy)^{p^i})$ is divisible by p^{i+1};
(iii) in view of the definition of g_i, it suffices to compute the trace of a power of an integral matrix modulo p^{i+1}. Using the fact that the exponent in question is at most n and that the entries of the matrix are from the interval $[0, p]$, the statement follows. □

With Theorems 2.5 and 2.6 we completely proved the properties (1)–(4) of the ideals I_i and functions g_i described at the beginning of this section. We have the following.

THEOREM 2.7. *Let A be an n-dimensional algebra over the field $GF(q)$ given by a collection of structure constants. Then (a basis of) $Rad(A)$ can be computed in time polynomial in n and* $\log q$. □

To conclude this section, we outline how finding the nilpotent and the solvable radical of a Lie algebra can be reduced to the associative case. First, we deal with the nilradical. We shall use a theorem of Jacobson (1962). Let L be a finite-dimensional Lie algebra over the field K.

JACOBSON'S THEOREM. *Let A denote the associative (matrix-) algebra generated by $ad(L)$, the adjoint representation of L. Then an element $x \in L$ is in the nilradical $N(L)$ if and only if $ad(x) \in Rad(A)$.*

This result settles the problem of computing $N(L)$ if K is a finite field or an algebraic number field. Indeed, we can compute a basis of A, and using the results of this section we can compute $Rad(A)$ as well. By solving a system of linear equations we can compute the intersection of the subspaces $ad(L)$ and $Rad(A)$.

COROLLARY 2.8. *Let L be a finite dimensional Lie algebra over the field K, where K is either finite or an algebraic number field, given by a collection of structure constants. Then the nilradical $N(L)$ can be computed in time polynomial in the input size.* □

Now we turn to the problem of computing the solvable radical $R(L)$. Over fields of characteristic zero, Beck, Kolman & Steward (1977) have given an efficient algorithm to compute $R(L)$. Their method is based on a characterisation of $R(L)$ using the Killing form, which is similar to Dickson's theorem. This characterisation (as well as Dickson's theorem) breaks down over fields of positive characteristic.

If K is finite then the problem of computing $R(L)$ can be, at least in a theoretical sense, efficiently reduced to the problem of computing $N(L)$. Indeed, we have $N(L) \leq R(L)$ and if $N(L) = (0)$ then $R(L) = (0)$, because the next to last element of the derived series of $R(L)$ is an abelian, therefore nilpotent ideal of L.

One can define the sequence L_i as follows: let $L_0 = L$ and if $N(L_i) \neq (0)$ then let $L_{i+1} = L_i/N(L_i)$, otherwise L_{i+1} is not defined. This sequence of Lie algebras has at most $dim_K L + 1$ elements. By Corollary 2.8, these algebras can be computed in polynomial time if K is finite. If L_j is the last element of the sequence then $L_j \cong L/R(L)$ and we can easily produce a basis for $R(L)$ by keeping track of the preimages of the ideals we factored out during the process.

COROLLARY 2.9. *Let L be an n dimensional Lie algebra over $GF(q)$, given by structure constants. Then (a basis of) the solvable radical $R(L)$ can be computed in time polynomial in n and* $\log q$. □

3. Finding the Simple Components of Semisimple Algebras

Our aim here is to find the minimal ideals, i.e. the Wedderburn decomposition of a finite semisimple associative algebra. We describe a polynomial time f-algorithm to solve this problem. This method was given by Friedl & Rónyai (1985), based on Friedl (1983). The algorithm is presented here because it is a major building block of our subsequent methods. A minor improvement of the original version is also included.

The input of the problem is a finite-dimensional semisimple associative algebra A over the field $GF(q)$ ($q = p^s$, p prime), inputted as a collection of structure constants. By the Wedderburn structure theorem

$$A = A_1 \oplus A_2 \oplus \cdots \oplus A_k,$$

where $A_1, A_2, \ldots, A_k$ are the minimal ideals of A. We give an f-algorithm running in time polynomial in $dim_{GF(q)} A$ and $\log q$ to find bases for the ideals A_i.

The problem can be reduced in polynomial time to the case when A is commutative. One can consider the centre of A

$$Z(A) \stackrel{\text{def}}{=} \{x \in A;\ xy = yx \text{ for every } y \in A\}.$$

A basis of the algebra $Z(A)$ can be obtained by solving a system of linear equations over $GF(q)$, since $x \in Z(A)$ if and only if $xa_i = a_i x$, $i = 1, \ldots, n$ holds where $a_1, \ldots, a_n$ is a basis of A over $GF(q)$. $Z(A)$ is a semisimple associative algebra and its Wedderburn decomposition relates nicely to that of A:

$$Z(A) = Z(A_1) \oplus Z(A_2) \oplus \cdots \oplus Z(A_k).$$

Knowing the subalgebras $Z(A_i)$, we can easily compute the ideals A_i because of the relation $A_i = Z(A_i)A$ (which, in fact, shows that A_i is the ideal of A generated by $Z(A_i)$). A basis of the product $Z(A_i)A$ can be obtained by simply selecting a maximal linearly independent set from the set of all possible products $b_j a_r$ where b_j and a_r run through a basis if $Z(A_i)$ and A, respectively.

From now on we shall assume that A is a commutative semisimple algebra over $GF(q)$. The method is an iteration which goes sequentially through a basis $a_1, \ldots, a_n$ of A. It either finds a decomposition of A into a direct sum of two smaller ideals $A = I \oplus J$ or concludes that A is a field, therefore direct irreducible. In the former case this cutting procedure can be applied to the ideals I and J and so on, since they are also semisimple commutative algebras and their ideals are also ideals of A. Thus, a call of the cutting procedure either concludes that an ideal is minimal or obtains a finer direct decomposition of A.

The cutting procedure works as follows. We consider a basis $a_1, \ldots, a_n$ of A. After processing the elements $a_1, \ldots, a_i$, the loop invariant is that F_i, the subalgebra generated by $a_1, \ldots, a_i$ is a field ($F_0 = GF(q)$). If $i = n$ then we have proved that A is a field, therefore it has no proper ideals. If $i < n$ then we compute the minimal polynomial f of the element $b = a_{i+1}$ over the field F_i. This involves finding the first linear dependence over F_i of the elements $1_A, b, b^2, \ldots, b^n$. Next, we factor f into irreducible factors over F_i (this is where we have to call the factoring oracle). If f is irreducible, then $F_i(a_{i+1})$ is a field. In this case we put $F_{i+1} = F_i(a_{i+1})$ and the ith step is finished. If f is reducible then f can be written as $f = gh$, where g and h are nonconstant relatively prime polynomials because A is a direct sum of fields. Now we put $I = Ag(a_{i+1})$ and $J = Ah(a_{i+1})$. I and J are obviously proper ideals of A, and using the fact that g and h are relatively prime, one can easily see that $A = I \oplus J$.

We have the following result.

THEOREM 3.1 (Friedl & Rónyai, 1985). *Let A be a finite dimensional semisimple associative algebra over the field $GF(q)$ ($q = p^s$, p prime), given by a collection of structure constants. Then there exists an f-algorithm running in time polynomial in $\dim_{GF(q)} A$ and $\log q$ to find bases for the minimal ideals of A (i.e. to find the Wedderburn decomposition of A).*

By putting the polynomial time Las Vegas method (Berlekamp, 1970) and the deterministic method (Berlekamp, 1968) into the oracle for factoring polynomials over finite fields, we obtain the following.

COROLLARY 3.2. *The minimal ideals of the algebra A above can be found by a Las Vegas algorithm running in time polynomial in $\dim_{GF(q)} A$ and $\log q$. Similarly, the minimal ideals of A can be found by a deterministic method running in time polynomial in $\dim_{GF(q)} A$ and q.*

REMARKS.

1. With some modifications, the algorithm of Theorem 3.1 works over algebraic number fields as well. One has to control the sizes of the intermediate results obtained from the repeated calls of the cutting procedure (Friedl & Rónyai, 1985).

2. From a practical point of view, a modified version of the above method seems to be more promising. A is a semisimple commutative algebra, hence the minimal ideals A_i are fields. These ideals contain a copy of the prime field $GF(p)$. We infer that A contains, as a subalgebra, a direct sum of k copies of $GF(p)$. This subalgebra B is the set of the fixed points of the map $x \mapsto x^p$. This is a $GF(p)$ linear map, consequently B can be computed by

solving a system of linear equations over $GF(p)$. Again, it suffices to compute the decomposition of B, for the minimal ideals of B generate the minimal ideals of A. Now during the cutting procedure we always work over the prime field (i.e. F_i never grows) and the minimal polynomials split into linear factors in the prime field. Note that this reduction is essentially the same as Berlekamp's reduction (1968) of factoring polynomials to finding roots in the prime field.

4. Simple Algebras over Finite Fields

The main task of the cutting procedure in section 3 was to find *zero divisors* in the algebra A, i.e. nonzero elements $x, y \in A$ such that $xy = 0$. There we solved this problem in the special case when A was a semisimple but not simple algebra. Here we consider this algorithmic problem in a broader context.

We have a finite associative algebra A given as a collection of structure constants, find a pair of zero divisors in A if there are any.

We give a polynomial time f-algorithm to solve the problem. As it turns out, the most important special case is when A is simple. Our solution of this case is based on an almost constructive proof of Wedderburn's theorem on finite division algebras presented in Herstein (1968). We adapt the proof to full matrix algebras and replace the purely existential steps by constructive ones. Let $F = GF(q)$. First, we study the zero divisors of $M_n(F)$. Let $a \in M_n(F)$, $a \notin F$ such that $L = F(a)$ is a field. Let l denote the degree of L over F. In other words, the minimal polynomial f of a over F is irreducible over F and $deg(f) = l$.

LEMMA 4.1. *There exists a* $c \in M_n(F)$ *such that:*

(*i*) $c^{-1}ac = a^q$;
(*ii*) *if* $Alg(a, c)$ *denotes the F-algebra generated by* a *and* c *then* $Alg(a, c)$ *is not commutative;*
(*iii*) $Alg(a, c) = L + cL + \cdots + c^m L + \cdots$.

PROOF. L is a simple subalgebra of $M_n(F)$ and the automorphism of L sending a to a^q leaves F element-wise fixed; therefore, by a theorem of Noether and Skolem (Pierce, 1982, section 12.6) this automorphism is inner, showing the existence of a $c \in M_n(F)$ satisfying (i). The statements (ii) and (iii) are valid for any such c. □

The next statement helps us to simplify the search for zero divisors. Let c be an arbitrary element satisfying (i) above.

LEMMA 4.2. *If* $Alg(a, c) = M_n(F)$ *then* $l = n$, $dim_F F(c) = n$, $c^n \in F$ *and*

$$Alg(a, c) = L \oplus cL \oplus \cdots \oplus c^{n-1}L. \tag{4.1}$$

PROOF. Simple calculation shows that for an arbitrary natural number i we have $c^{-i}ac^i = a^{q^i}$. This implies that $ac^l = c^l a$, therefore c^l is in the centre of $Alg(a, c) = M_n(F)$, which is F. The element c satisfies a polynomial of degree l over F, so we have

$$Alg(a, c) = L + cL + \cdots + c^{l-1}L,$$

by Lemma 4.1. This implies that $dim_F Alg(a, c) \leq l^2$. We have equality here iff the sum is a direct sum. As l is the degree of the minimal polynomial of a, we have $l \leq n$. Now from $dim_F Alg(a, c) = n^2$ we obtain that $l = n$ and the sum must be a direct sum. Finally, c cannot

satisfy a polynomial over F with degree less than n for otherwise we had a decomposition shorter than (4.1), which is impossible. □

We continue the study of the case $Alg(a, c) = M_n(F)$. We have seen that c is a root of a polynomial of the form $x^n - \alpha$ where $\alpha \in F$. We know also that this is the minimal polynomial of c over F. The *norm* of an element d of L is defined as

$$norm(d) := dd^q d^{q^2} \cdots d^{q^{n-1}}.$$

(This is the L/F relative norm.) The next statement plays a key role in Herstein's proof of Wedderburn's theorem.

LEMMA 4.3. *Let $d \in L$ be an element such that $norm(d) = 1/\alpha$. Then $1 - cd$ is a zero divisor in the algebra $Alg(a, c) = M_n(F)$.*

PROOF. Let us define the element $z \in Alg(a, c)$ as

$$z = 1 + cd + c^2 dd^q + \cdots + c^{n-1} dd^q \cdots d^{q^{n-2}}.$$

A straightforward computation shows that $z(1 - cd) = 0$. On the other hand, the fact that (4.1) is a direct sum implies that neither z nor $1 - cd$ can be zero, proving the claim. □

We are unable to solve the above norm equation in general because of the high degree of the polynomial involved. To avoid this difficulty, we impose an additional restriction on c.

LEMMA 4.4. *Suppose that $Alg(a, c) = M_n(F)$ as above and that the minimal polynomial $x^n - \alpha$ of c is irreducible over F. Then $g(x) = x^n - 1/\alpha$ is also irreducible over F. Moreover, g splits into linear factors in L and if n is odd then $d \in L$, $g(d) = 0$ imply that $norm(d) = 1/\alpha$.*

PROOF. The irreducibility of $x^n - \alpha$ means that $F(c)$ is a field and $dim_F F(c) = n$. It is clear that $F(c) = F(1/c)$, therefore g is also irreducible over F. By comparing the degrees of the fields L and $F(c)$ we see that $F(c) \cong L$, so g splits into linear factors in L. As for the last statement, let d be an arbitrary root of g from L. The irreducibility of g implies that its constant term can be written as $(-1)^n norm(d) = -norm(d) = -1/\alpha$, giving the last statement. □

The proof of Lemma 4.4 covers the first part of the following lemma.

LEMMA 4.5. *Suppose that either n is odd or $n = 2$ and let L be an extension field of F such that $dim_F L = n$ and let $g(x) = x^n - \beta$, $\beta \in F$ be a given irreducible polynomial over F. Then we can find an element $d \in L$ such that $norm(d) = \beta$ by a polynomial time f-algorithm.*

PROOF. It suffices to consider the case $n = 2$. We distinguish two cases.

Case 1. $-\beta$ is a quadratic nonresidue in F. If $d \in L$ is a root of $h(x) = x^2 + \beta$ then the other root must be d^q and calculating the constant term of g we obtain that $d^{q+1} = norm(d) = \beta$.

Case 2. $-\beta$ is a quadratic residue in F. Let γ be an element of F such that $\gamma^2 = -\beta$. If we can find an element $b \in L$ such that $norm(b) = -1$ then by letting $d = \gamma b$ we obtain $norm(d) = \gamma^2 norm(b) = \beta$.

To solve the norm equation $norm(b) = -1$, it suffices to find a generating element b of the Sylow 2-subgroup of the multiplicative group L^* of L. Such an element b can be found by solving at most $2 \log_2 q$ quadratic equations in L. To prove that b is a solution, we remark that if b^{q+1} were a residue in F, then it would imply that

$$b^{(q-1)(q+1)/2} = 1,$$

contradicting the fact that b is a nonresidue in L. The element $norm(b)$ is in the Sylow 2-subgroup of F^*, therefore $norm(b) = -1$.

In all cases the norm equation can be solved by factoring moderately sized polynomials. The proof is complete. □

Now we are ready to describe our algorithm to find zero divisors in finite algebras. Suppose that we have an algebra A over the field $Z = GF(p^r)$, p prime and $dim_Z A = m$. The algebra A is given by structure constants. Our objective is to find a pair of zero divisors in A. The procedure ZERODIV( ) returns a pair of zero divisors x, y of A if there are any.

procedure ZERODIV(A)

Step 1. *Compute $Rad(A)$ using the method of section 2. If $Rad(A) \neq (0)$ then pick an arbitrary nonzero element $x \in Rad(A)$. As x is nilpotent, an appropriate power of it will suffice as y,* **return**(x, y).

Step 2. (**A is semisimple**)
Determine the Wedderburn decomposition of A, using the method of section 3. If A is not simple, say $A = I \oplus J$ where I and J are nonzero ideals, then x and y can be arbitrary nonzero elements of I and J, respectively.
return(x, y).

Step 3. (**A is simple**)
Check whether A is commutative. In case of an affirmative answer terminate concluding that A is a field (and therefore it has no zero divisors).

Step 4. (**A is a full matrix algebra over some field extension $F = GF(q)$ of Z, say $A \cong M_n(F)$ and $n > 1$.**)
Pick an arbitrary element $b \in A$ such that $b \notin F$. Next, compute and factor the minimal polynomial f of b over F. If f is reducible over F, say $f = gh$ a proper factorisation then **return**$(g(b), h(b))$.

Step 5. (**f is irreducible, therefore $F(b)$ is a field.**)
If $dim_F F(b)$ is even then choose an $a \in F(b)$ such that $dim_F F(a) = 2$ (we have to find a solution of the system of linear equations $a^{q^2} = a$ which is not in F), otherwise let $a := b$.

Step 6. (**$F(a)$ is a field and $l = dim_F F(a)$ is either odd or it is 2.**)
Find a nonzero element $c \in A$ such that $ac = ca^q$ (by solving a system of linear equations). Compute and factor the minimal polynomial of c over F. If it is reducible over F then **return** *zero divisors as in step* 4.

Step 7. (**c is an invertible element of A and $c^{-1}ac = a^q$ holds.**)
Form $Alg(a, c)$, the F-algebra generated by a and c. If $Alg(a, c) \neq A$ then let $A := Alg(a, c)$ and **go back to** *step* 1.

Step 8. (**We have here* $n = l$, $Alg(a, c) = A \cong M_n(F)$, *n is either odd or* $n = 2$, *the minimal polynomial of c over F is* $f(a) = x^n - \alpha$ *for some* $\alpha \in F$ *and f is irreducible over F.**)
Find a solution $d \in F(a)$ *of the norm equation* $norm(d) = 1/\alpha$ *using the algorithm of Lemma* 4.5.
Now put $x := 1 - cd$ *and*

$$y := 1 + cd + c^2 dd^q + \cdots + c^{n-1} dd^q \cdots d^{q^{n-2}}.$$

return(x, y).

end procedure

THEOREM 4.6. *Let A be an algebra over* $GF(p^r)$, $dim_{GF(p^r)} A = m$. *The procedure ZERODIV( ) finds a pair of zero divisors in A if A contains zero divisors. As an f-algorithm it runs in time polynomial in m, r and* $\log p$.

PROOF. The correctness and the timing of steps 1–3 is covered in Theorems 2.7 and 3.1. The procedure is essentially an iteration. If we enter the loop, i.e. we go back to step 1 from step 7, then *the actual algebra A is not commutative and thus contains zero divisors* by Lemma 4.1(ii). This statement may serve as a loop invariant. It is also clear that the dimension of A strictly decreases during an iteration step, except the last one. If we terminate at steps 4 or 6 then we clearly found zero divisors. The correctness of the annotation of step 8 follows from Lemma 4.2 and from the remark after it, so we can apply Lemma 4.3 to show that x and y form, indeed, a pair of zero divisors. We can solve the norm equation using the method of Lemma 4.5.

The structure of the loop shows that a step is executed at most m times. This implies a polynomial bound on the running time, provided that we have polynomial bounds on the time required by the individual steps. For steps 1–3 this follows from Theorems 2.7 and 3.1.

In steps 4–8 the major computational tasks are: solving systems of linear equations with at most m equations and at most m unknowns over F (or over Z); factoring polynomials of degree at most m over F; computing a (basis of a) subalgebra generated by two elements. These clearly can be done in time polynomial in the input size if we use an oracle for factoring polynomials. The norm equation from step 8 can be solved in polynomial time by Lemma 4.5 and the element y can then be computed efficiently using fast exponentiation. The proof is complete. □

For factoring polynomials over finite fields, we can use either the Las Vegas method of Berlekamp (1970), Rabin (1980) and Cantor & Zassenhaus (1981), or the deterministic method of Berlekamp (1968).

COROLLARY 4.7. *The problem of finding zero divisors in A can be solved by a Las Vegas method running in time polynomial in m, r and* $\log p$. *Also, there exists a deterministic algorithm running in time polynomial in m, r and p.* □

5. Applications

We give three applications of the results to some algorithmic problems from computational algebra. Applications to special types of algebras will be considered elsewhere.

5.1. EXPLICIT ISOMORPHISMS OF MATRIX ALGEBRAS

From the results of sections 2 and 3 we can easily see that there exists an efficient f-algorithm to decide if a given finite algebra A is isomorphic to a full matrix algebra. Indeed, it suffices to check if $Rad(A) = (0)$ and whether A is directly indecomposible. In case of an affirmative answer, say if $A \cong M_n(F)$, then we can also find n and F. The problem remains to establish an explicit isomorphism from A to $M_n(F)$ (i.e. to represent A as an algebra of linear transformations of an n-dimensional linear space V over F). To find such a representation, it suffices to find an idempotent $e \in A$ such that $rank(e) = 1$ (here e is viewed as an element of $M_n(F)$; in the light of Lemma 5.1 the rank is independent from the actual isomorphism). Indeed, we can put $V = M_n(F)e$. It is well known that $dim_F V = n$ and $M_n(F)$ acts nontrivially and therefore faithfully on V via multiplication from the left.

The following easy lemma will be useful. The straightforward proof is omitted.

LEMMA 5.1. *Let $e \in M_n(F)$ be an idempotent such that $rank(e) = m$. Then $eM_n(F)e \cong M_m(F)$.*

This lemma shows that it is enough to give an algorithm to find a *singular* idempotent in A: if $e \in M_n(F)$, $rank(e) = m < n$ then we reduced the problem to the smaller instance $eAe \cong M_m(F)$. Indeed, if f is a rank one idempotent of eAe, then f is a rank one idempotent in A as well, because e is the identity element of eAe and this implies that $F \cong feAef = fAf$.

To obtain a singular idempotent, we call ZERODIV(A). If $n > 1$ then it returns a zero divisor x. Let e be the right identity element of the left ideal Ax. The element e is obviously a singular idempotent and can be found by solving a system of linear equations (describing that e is the identity element of Ax), once x is given. We have the following

THEOREM 5.2. *Suppose that we have an algebra A such that $A \cong M_n(GF(q))$. Then an explicit isomorphism from A to $M_n(GF(q))$ can be constructed by an f-algorithm running in time polynomial in n and* $\log q$. □

An explicit isomorphism is useful for the usual representation of $M_n(F)$ (as the algebra of all n by n matrices over F) is easy to handle. For example, we can conveniently decompose it into a direct sum of minimal left ideals. If e_{ii} denotes the matrix having 1 in position (i, i) and zeros elsewhere, then we have

$$M_n(F) = M_n(F)e_{11} \oplus M_n(F)e_{22} \oplus \cdots \oplus M_n(F)e_{nn}.$$

It is easy to check that $M_n(F)e_{ii}$ is a minimal left ideal $i = 1, \ldots, n$. If we have an explicit isomorphism from A to $M_n(F)$ then we can obtain a decomposition of A. This observation can be generalised to an arbitrary finite semisimple algebra B. Using the techniques of section 3, B can be decomposed into a direct sum of minimal ideals. The minimal ideals are simple algebras, hence we can decompose them into a direct sum of minimal left ideals using the above method. By putting these minimal left ideals together, we obtain a decomposition of B.

COROLLARY 5.3. *Let B be a semisimple algebra over $F = GF(q)$, $dim_F B = m$. B can be decomposed into a direct sum of minimal left ideals by an f-algorithm running in time polynomial in m and* $\log q$.

5.2. COMMON INVARIANT SUBSPACES

Now we apply our methods to solve an important linear algebraic problem over a finite ground field. Let $X_1, \ldots, X_k \in M_n(F)$ and consider their action on the linear space V of n by 1 column vectors over F. We would like to find a nontrivial subspace $U \leq V$ such that $X_i U \leq U$ for $i = 1, \ldots, k$, if there is any. This problem is solved by the procedure INV( ). It has one input parameter. It must be a nonempty set $S \subset M_n(F)$. For the sake of simplicity we also assume that the zero matrix is not in S. It outputs a proper invariant subspace if there is such a subspace.

procedure INV(S)

Step 1. *Compute A, the algebra of matrices generated by S (i.e. compute a basis of A over F).* (**S and A have the same invariant subspaces.**)

Step 2. *If $AV < V$ then* **return**(AV).

Step 3. *Compute $Rad(A)$, the radical of A. If $Rad(A) > (0)$ then* **return**($Rad(A)V$).

Step 4. (**A is semisimple and V is a unitary A-module.**)
Break A into a direct sum of minimal left ideals:

$$A = \rho_1 \oplus \cdots \oplus \rho_m.$$

Let v be an arbitrary nonzero vector from V and consider the (A-invariant) subspaces $\rho_1 v, \ldots, \rho_m v$ and let U be any of these which is not (0). If $U = V$ then there is no nontrivial invariant subspace, otherwise **return**(U).

end procedure

First, we look at the correctness of INV( ). It is obvious that S and A have the same invariant subspaces. If we terminate at step 2 then we have a nontrivial invariant subspace because of the relation $AV > (0)$. If we terminate at step 3 then $Rad(A)V$ is an A-invariant subspace. This subspace is different from (0), for $M_n(F)$ acts faithfully on V. As $Rad(A)$ is a nilpotent algebra, the equality $V = Rad(A)V$ is impossible, hence we have a nontrivial invariant subspace. If we are at step 4 then A is semisimple. It is known (cf. Herstein, 1968, pp. 97–98) that $\rho_i v$ is either (0) or a minimal A-invariant subspace. The fact that we have survived the test of step 2 implies that V is a unitary A-module (i.e. the identity element of A is the identity matrix). This implies that not all of the subspaces $\rho_i v$ can be (0). We conclude that U is a minimal invariant subspace. In particular, if $U = V$, then V has no proper invariant subspaces. The correctness is proved.

Steps 1 and 2 can obviously be done in polynomial time. As for step 3, we have a polynomial time algorithm to compute $Rad(A)$, cf. Theorem 2.7. By Corollary 5.3 the last step can also be done by a polynomial time f-algorithm. We have the following corollary.

COROLLARY 5.4. *Let $S = \{X_1, \ldots, X_k\} \subset M_n(F)$. A nontrivial invariant subspace can be found (if there is any) by an f-algorithm running in time polynomial in k, n and $\log q$. In particular, the problem can be solved by a Las Vegas method running in time polynomial in the parameters n, k, $\log q$ and by a deterministic method running in time polynomial in the parameters n, k, p and r, where $q = p^r$.* □

REMARK. The above methods can be extended to obtain an efficient method to solve a fundamental problem of representation theory: to decompose finite modules over finite

semisimple algebras into a direct sum of simple submodules. In fact, if A is semisimple, then INV( ) returns a simple submodule of V and if we can find simple submodules, then we can find a direct decomposition using standard methods (cf. Herstein, 1968, pp. 97–98).

5.3. AN APPLICATION TO PERMUTATION GROUPS

In this section we give a deterministic polynomial time method to solve an algorithmic problem related to permutation groups; to find a minimal normal subgroup in an elementary abelian interval of permutation groups. More precisely, consider the following setting. Let $G \leq S_n$ and let $K < H$ normal subgroups of G such that H/K is an elementary abelian p-group for some prime p. Suppose that G, H, K are given by generating sets. Let m denote the number of generating elements we have. Our aim is to find a minimal normal subgroup L of G such that $K < L \leq H$.

First, we compute strong generating sets (cf. Luks, 1982) of the above groups. These have $O(n^2)$ elements. The factor group $V := H/K$ can be considered as a linear space of dimension $< n$ over $GF(p)$. The elements of G act on V as $GF(p)$-linear transformations via conjugation. Our problem is equivalent to finding a minimal G-invariant subspace of V. Obviously, it suffices to find a minimal Φ-invariant subspace, where Φ is the generating set of G we have. First, we compute a basis of V. This task can be done in polynomial time. Next, we compute matrix representations of the linear transformations corresponding to the elements of Φ. We can then apply INV( ) to find a minimal invariant subspace. As p is small, we can use the deterministic factoring method of Berlekamp (1968).

COROLLARY 5.4. *The above problem can be solved by a deterministic algorithm running in time polynomial in n and m.*

The author is indebted to L. Babai for suggesting the main problems considered here and for his constant interest and helpful comments. Helpful conversations on the subject with S. Becker, K. Friedl, W. M. Kantor and E. M. Luks are gratefully acknowledged. Thanks are due to the referee for helpful suggestions.

References

Babai, L. (1979). Monte-Carlo algorithms in graph isomorphism testing. *Tech. Rep. 79–10*, Department of Mathematics and Statistics, University of Montreal.

Berlekamp, E. R. (1968). *Algebraic Coding Theory*. New York: McGraw-Hill.

Berlekamp, E. R. (1970). Factoring polynomials over large finite fields. *Math. of Computation* **24**, 713–715.

Beck, R. E., Kolman, B., Stewart, I. N. (1977). Computing the structure of a Lie algebra. In: *Computers in Nonassociative Rings and Algebras*, pp. 167–188. London: Academic Press.

Ben-Or, M. (1981). Probabilistic algorithms over finite fields. *Proc. 22th IEEE FOCS*, 394–398.

Camion, P. (1983). A deterministic algorithm for factorizing polynomials of $F_q[x]$. *Annals of Discrete Math.* **17**, 149–157.

Cantor, D. G., Zassenhaus, H. (1981). A new algorithm for factoring polynomials over finite fields. *Math. of Computation* **36**, 587–592.

Dickson, L. E. (1923). *Algebras and Their Arithmetics*. Chicago, IL: University of Chicago Press.

Friedl, K. (1983). *Algorithms in Algebra* (in Hungarian). Diploma Thesis. Eötvös University, Budapest.

Friedl, K., Rónyai, L. (1985). Polynomial time solutions of some problems in computational algebra. *Proc. 17th ACM STOC*, pp. 153–162. Providence, RI.

Herstein, I. N. (1968). *Noncommutative Rings*. Mathematical Association of America.

Humphreys, J. E. (1980). *Introduction to Lie algebra and representation theory. GTM* **9**. New York: Springer.

Hopcroft, J. E., Ullman, J. D. (1979). *Introduction to Automata Theory, Languages and Computation*. Reading, MA: Addison-Wesley.

Jacobson, N. (1962). *Lie Algebras*. New York: John Wiley.

Knuth, D. E. (1981). *The Art of Computer Programming, Vol. 2. Seminumerical Algorithms.* Reading, MA: Addison-Wesley.
Kertész, A. (1987). *Lectures on Artinian Rings* (ed. R. Wiegandt). Budapest: Akadémiai Kiadó.
Lidl, R., Niederreiter, H. (1983). *Finite Fields.* Reading, MA: Addison-Wesley.
Luks, E. M. (1982). Isomorphism of graphs of bounded valence can be tested in polynomial time. *J. Comput. Syst. Sci.* **25**, 42–65.
Pierce, R. S. (1982). *Associative Algebras.* Berlin: Springer.
Rabin, M. O. (1980). Probabilistic algorithms in finite fields. *SIAM J. Comp.* **9**, 273–280.